

ECE 4250 Digital Signal and Image Processing

Spring 2019

Project Information Sheet

March 19, 2019

1 Overview

In this project, you will develop an algorithm that will automatically segment a brain MRI scan into anatomical regions of interest. You will have access to some brain MRIs that have been manually segmented. There will be three milestones and your deliverables for each of these milestones are described below. Also, there will be an in-class competition at the end of the semester. You will be able to earn some bonus points depending on your performance in this competition. You will submit your work as an individual. You are encouraged to collaborate with your peers, however, the submitted work should be your own code and write-up. General guidelines regarding programming assignments apply.

2 In-class Competition

- At the end of the semester, you will submit your segmentation results on the test subjects as part of a competition, which we will evaluate and use to quantify performance (via Jaccard Index, see below).
- A non-trivial submission (see below) will be considered a valid submission.
- We will rank all valid submissions based on performance (from high to low).
- Top ranking individual(s) will receive a bonus grade of 5% total.
- The bottom ranking individual(s) will receive no bonus grade.
- All intermediate ranking individuals will receive a non-zero bonus grade up to 5% that will be linearly proportional to their performance.

Task	Total Weight on Final Grade
Milestone 1	5%
Milestone 2	10%
Final Report	15%
Valid Competition Submission	5%
Competition Bonus	5%

3 Project grade breakdown

4 Dataset

- You are given T1-weighted brain MRI volumes
- In total there are 18 subjects (6 training + 2 validation + 10 testing)
- Each subject has a single MRI scan
- These scans have been manually segmented into anatomical regions of interests
- You will receive the manual segmentations for the training and validations subjects
- All data are provided in ANALYZE format:
<http://imaging.mrc-cbu.cam.ac.uk/imaging/FormatAnalyze>
- You may use “nibabel” for Python to read the files

5 Project Milestones

- **ASAP:** Download all training and validation data:
https://drive.google.com/open?id=1DtglCdjRRFHSMBsPHvG_ut1CnhwtRJm4
Make sure to take a look at the README file.
- **Milestone 1:** You will submit a Jupyter notebook (Python 3) that implements following instructions:
 1. Load the MRI volumes
 2. Determine the pixel spacing and slice thickness of each loaded volume
 3. Extract, visualize, and save middle coronal slices for all training+validation cases, including the MRIs and segmentations
(Hint: <https://faculty.washington.edu/chudler/slice.html>)
- **Milestone 2:** You will submit a Jupyter notebook that implements following instructions. You are allowed to use built-in or already-available functions for spatial transformation and optimization.

1. Write a function that computes a 4-parameter geometric registration (global scale, rotation, and translations along two axes) between two mid-coronal MRI slices from two different subjects (a fixed image and a moving image). This step will consist of following sub-steps.
 - a function that takes in an input (moving) image, 4 transformation parameters (global scale, rotation, and translations along two axes) and output (fixed image) grid size, and computes the output (moved) image. The output image has a size of the output grid size and pixel values are obtained by resampling the input image onto the output grid via applying the geometric transformation: a global scale multiplied with a rotation matrix, followed by a translation. Outside-of-grid-range values can default to zero.
 - a loss function that takes three inputs: a length-4 vector of geometric transformation parameters, a fixed image, and a moving image. The output should be equal to the sum of squared differences between the geometrically transformed moved image and the fixed image.
 - an optimization module that minimizes the loss function for a given input image pair (fixed and moving). This module should return two things: the transformed moving image and the optimal geometric transformation parameters.
 2. Use your registration tool to resample each training image (moving) onto each validation image (fixed) (i.e., you need to run 12 registration instances). Visualize some slices of these results. You need to show that your registration works - i.e., plot results for before the registration and after the registration.
 3. Apply the registration results (optimal transformations) to resample the manual segmentations of each training subject onto the validation subject grids (use nearest neighbor interpolation)
 4. For every pixel on validation subject grid, compute the most frequent training label – this is called majority voting based label fusion. You can implement any tie-break strategy you want. This is a crude segmentation of the validation subjects.
 5. Write a function that computes the Jaccard overlap index for a given region of interest (ROI) between an input manual segmentation and an automatic segmentation. The Jaccard index is defined as the ratio between the area of the intersection and the area of the union, where the intersection and union are defined with respect to the manual segmentation and an automatic segmentation.
 6. Compute the Jaccard index for your automatic validation subject segmentations. Compile these in a table and print. Only consider following regions of interest (both left and right): Cerebral-White-Matter and Cerebral-Cortex.
- **Final Report and Competition:** Implement a more sophisticated label fusion based segmentation strategy to yield better segmentations. Feel free to optimize your approach using the validation subjects and their manual segmentations. You can explore various directions, including but not limited to:

1. Using an affine or non-linear transformation model that achieves better alignment than geometric transformations
2. Computing a weighted fusion approach where the training subjects (atlases) are weighed differently based on similarity between intensity values
3. A patch based approach that seeks similar atlas patches in a certain neighborhood
4. Replacing nearest neighbor interpolation with a different method

Next, download all the test subject images. Extract and save the mid-coronal slices. Once you settle on your best label fusion approach, you should use it to compute automatic segmentations of all the test subject mid-coronal slices. You will submit your test subject segmentations to us, which we will compute the Jaccard index values on. Your overall performance will be computed as the average Jaccard index across all subjects and following ROIs: Cerebral-White-Matter and Cerebral-Cortex. We will rank all submissions based on this performance metric. A performance metric greater than 50% will be considered as a valid submission.

In addition to a Jupyter Notebook that implements above, you will also submit a 6 page write-up (final report) that describes your effort/approach and illustrates some results.

6 Dates (all by 11:59pm)

- Milestone 1: due 3/29.
- Milestone 2: due 4/26.
- In-class competition submissions: due 5/13
- Final Report and final Jupyter notebook: due 5/15